

PATENT



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: **Davia et al.**

Serial No. 09/867,883

Filed: May 30, 2001

**For: Method and Computer Program
Product for Testing Application
Program Software**

§
§ Group Art Unit: 2122

§ Examiner: Vo, Ted T.

~~~~~

**Commissioner for Patents**  
**P.O. Box 1450**  
**Alexandria, VA 22313-1450**

## ATTENTION: Board of Patent Appeals and Interferences

**Certificate of Mailing Under 37 C.F.R. § 1.8(a)**

I hereby certify this correspondence is being deposited with the United States Postal Service as First Class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on July 26, 2004.

By: Michele Morrow  
Michele Morrow

**APPELLANT'S BRIEF (37 C.F.R. 1.192)**

This brief is in furtherance of the Notice of Appeal, filed in this case on May 26, 2004.

The fees required under § 1.17(c), and any required petition for extension of time for filing this brief and fees therefore, are dealt with in the accompanying TRANSMITTAL OF APPEAL BRIEF.

This brief is transmitted in triplicate. (37 C.F.R. 1.192(a))

07/30/2004 MHEKONEN 00000010 090457 09867883

01 FC:1402 330,00 DA

### **REAL PARTIES IN INTEREST**

The real party in interest in this appeal is the following party: International Business Machines Corporation.

### **RELATED APPEALS AND INTERFERENCES**

With respect to other appeals or interference's that will directly affect, or be directly affected by, or have a bearing on the Board's decision in the pending appeal, there are no such appeals or interference's.

### **STATUS OF CLAIMS**

#### **A. TOTAL NUMBER OF CLAIMS IN APPLICATION**

Claims in the application are: 1-16.

#### **B. STATUS OF ALL THE CLAIMS IN APPLICATION**

1. Claims canceled: NONE
2. Claims withdrawn from consideration but not canceled: NONE
3. Claims pending: 1-16
4. Claims allowed: NONE
5. Claims rejected: 1-16

#### **C. CLAIMS ON APPEAL**

The claims on appeal are: 1-16.

## **STATUS OF AMENDMENTS**

There are no amendments after final rejection.

## **SUMMARY OF INVENTION**

The present invention provides a way of testing application programs efficiently and effectively across a variety of operating systems, processors, and configurations. (Specification, page 6, lines 3-5) A test bed according to the present invention includes a primary test computer, which may be operated by a technician and a number of secondary test computers, which also execute the application under test. (Specification, page 7, line 10 to page 8, line 3) The processors, operating system, configurations and so forth of the various secondary test computers may differ among each other and may differ from the processor, operating system, or configuration of the primary test computer. (Specification page 7, lines 4-8)

## **ISSUES**

The only issue on appeal is whether claims 1-16 are anticipated under 35 U.S.C. § 102(e) based on BEA, “BEA MessageQ – Programming Guide” (3-2000).

## **GROUPING OF CLAIMS**

The claims stand or fall together as a single group.

## **ARGUMENT**

The Final Office Action rejects claims 1-16 under 35 U.S.C. § 102(e) as being anticipated by BEA, “BEA MessageQ – Programming Guide” (3-2000). This rejection is respectfully traversed.

Appellants respectfully submits that, contrary to the allegations made in the Final Office Action, the BEA reference does not, in fact, teach sending the input data from the primary test computer to a secondary test computer also testing the application program, receiving the input data by the secondary test computer and executing the application program by the secondary test computer responsive to the input data, as recited in independent claims 1, 10 and 14.

BEA is directed to Open VMS programming for an easy-to-use, fast, and reliable message software that allows applications to communicate using the industry-leading queued message bus technology. Chapter 2 of BEA is directed to applications that send messages using the BEA MessageQ functions and one of two types of delivery modes: recoverable or nonrecoverable. If a message is sent as nonrecoverable, the message is lost if it cannot be delivered to the target queue unless the application incorporates an error recovery procedure. If the message is sent as recoverable, BEA MessageQ Message Recovery Services (MRS) automatically guarantees delivery to the target queue in spite of system, process, and network failures. Chapter 7 of BEA is directed to using the BEA MessageQ Script Facility to provide a powerful tool for application developers to use in simulating message exchange between programs. Instead of writing a test program, BEA creates a script file containing instructions for capturing messages sent or received by an application, replaying captured messages, or simulating messages sent from an application that is still under development.

### **I. 35 U.S.C. § 102, Alleged Anticipation, Claims 1-16**

Claim 1, which is representative of the other rejected independent claims 10 and 14 with regard to similarly recited subject matter, reads as follows:

1. A method for testing an application program, comprising the steps of:  
capturing input data entered into a primary test computer testing an application program;  
sending the input data from the primary test computer to a secondary test

computer also testing the application program;  
receiving the input data by the secondary test computer;  
executing the application program by the secondary test computer  
responsive to the input data;  
when an exception condition occurs while executing the application  
program by the secondary computer, reporting the exception condition to the  
primary test computer immediately upon detection of the exception condition; and  
displaying the exception condition by the primary test computer.

Appellants respectfully submit that BEA does not identically show each and every feature of the claims arranged as they are in the claims. Specifically, BEA does not teach sending the input data from the primary test computer to a secondary test computer also testing the application program, receiving the input data by the secondary test computer and executing the application program by the secondary test computer responsive to the input data. BEA teaches a platform where only applications that send messages are described. Moreover, the BEA reference only simulates the message exchange between programs and not testing of the application program, as shown on page 7-1, first paragraph, which reads as follows:

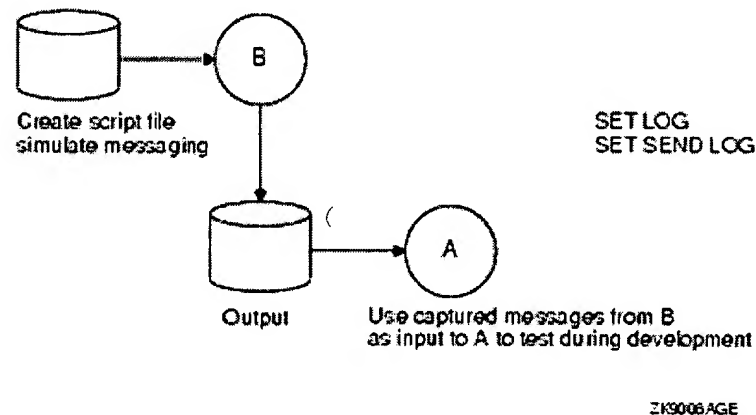
The BEA MessageQ Script Facility provides a powerful tool for application developers to use in simulating message exchange between programs. Instead of writing a test program, you create a script file containing instructions for capturing messages sent or received by an application, replaying captured messages, or simulating messages sent from an application that is still under development.

In response to the argument that BEA does not teach sending the input data from the primary test computer to a secondary test computer also testing the application program, receiving the input data by the secondary test computer and executing the application program by the secondary test computer responsive to the input data, the Advisory Action states:

Examiner would respond that the whole BEA teaches the broad recitation of the claim. The reference's applicant A and application B have means of Computer A and Computer B because BEA refers then to Client/server (Page 7-1, the fourth indentation). Application A and Application B stand for the testing computers. Relatively, if regards Application B as a primary test computer and A as a secondary test computer then Figure 7-6 is showing A testing the input from B (See page 7-8, "Capture message from Scripts"). Figure 7-6 shows "Creating script file simulating messaging" pointed to B; this is an input to B, the script file stands for the testing program, and A test the output (that is also input to B) sending from B. Examiner also refers to page 7-8: Capture messages using Scripts. It is noted that the script file is created in B as shown in Figure 7-6. The

words “Capture messages using Scripts” disclose “capturing input data entered from the primary test computer”. In the section “Capturing messages using Scripts” it shows “SET RECEIVE”; this clearly shows a message is captured at B by this command and sent to A for testing.

The present invention is directed to testing application program software. While BEA may mention the terms client and server, there is nothing in any section of BEA that states the server is on one computer and the client is on another computer. Furthermore, BEA does not teach sending the input data from the primary test computer to a secondary test computer also testing the same application program that the primary test computer is testing. The Office Action alleges that this feature is taught at Figure 7-6, elements A and B, shown as follows:



BEA MessageQ Programmer’s Guide, Figure 7-6. In this Figure, BEA teaches that Application B receives a script file for testing and the output of Application B is sent to Application A for testing which captures unsent messages. There is nothing in this section, or any other section of BEA, that teaches capturing input data entered into a primary test computer testing an application program and sending the input data from the primary test computer to a secondary test computer also testing the application program. The reference does not state that Application B, which is running on a primary computer, is the same as Application A, which is running on a secondary computer. Application A and Application B are presumably different applications. This presumption is supported by the Examiner’s statement that BEA “clearly shows a message is captured at B by this command and sent to A for testing,” (Advisory Action Continuation Sheet). Thus, Application B is a capturing application that captures messages and then these messages

are sent to Application A for testing. Application A is an actual simulated test computer application, which is running on the same computer as Application B. This is different from the present invention where input data entered into a primary test computer testing an application program is captured and sent from the primary test computer to a secondary test computer also testing the same application program.

Still further, BEA does not teach receiving the input data by the secondary test computer. The Office Action alleges that this feature is taught in Figure 7-6, shown above. As shown above, the data received by Application A is the output of Application B and is not the captured input data entered into a primary test computer testing an application program. Thus, BEA teaches receiving the output data by Application B and does not teach receiving the input data by the secondary test computer, which is the same input data captured into a primary test computer.

Even further, BEA does not teach executing the application program by the secondary test computer responsive to the input data. While BEA executes Application A in response to the data received from Application B, the data received is the output of Application B and not the input data captured into a primary test computer.

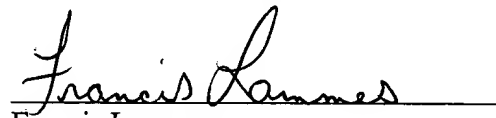
Thus, in view of the above, Appellants respectfully submit that BEA does not teach each and every feature of independent claim 1 as is required under 35 U.S.C. § 102(b). Similar distinctions apply to similar features found in independent claims 10 and 14. At least by virtue of their dependency on independent claims 1, 10 and 14, the specific features of dependent claims 2-9, 11-13 and 15-16 are not taught by BEA. Accordingly, Appellants respectfully submit that the rejection of claims 1-16 under 35 U.S.C. § 102(b) should be overturned.



## CONCLUSION

In view of the above, Appellants respectfully submit that claims 1-16 are allowable over the cited prior art and that the application is in condition for allowance. Accordingly, Appellants respectfully request the Board of Patent Appeals and Interferences to not sustain the rejections set forth in the Final Office Action.

Respectfully submitted,

A handwritten signature in cursive script, appearing to read "Francis Lammes", is written over a horizontal line.

Francis Lammes

Reg. No. 55,353

**YEE & ASSOCIATES, P.C.**

PO Box 802333

Dallas, TX 75380

(972) 367-2001

## **APPENDIX OF CLAIMS**

The text of the claims involved in the appeal are:

1. A method for testing an application program, comprising the steps of:  
capturing input data entered into a primary test computer testing an application program;  
sending the input data from the primary test computer to a secondary test computer also testing the application program;  
receiving the input data by the secondary test computer;  
executing the application program by the secondary test computer responsive to the input data;  
when an exception condition occurs while executing the application program by the secondary computer, reporting the exception condition to the primary test computer immediately upon detection of the exception condition; and  
displaying the exception condition by the primary test computer.
2. The method of claim 1, further comprising the step of logging the input data by the primary test computer.
3. The method of claim 1, further comprising the step of mapping the input data to a format compatible with the secondary test computer, wherein the step of mapping is performed by the secondary test computer.

4. The method of claim 1, further comprising the step of logging the exception condition, wherein the step of logging is performed by the primary test computer.
5. The method of claim 1, wherein the input data is entered using a keyboard.
6. The method of claim 1, wherein the input data is entered using a computer mouse.
7. The method of claim 1, wherein the primary test computer sends the input data to the secondary test computer over the Internet, and the secondary test computer reports the exception condition to the primary test computer over the Internet.
8. The method of claim 1, wherein the primary test computer sends the input data to the secondary test computer over an Intranet, and the secondary test computer reports the exception condition to the primary test computer over the Intranet.
9. The method of claim 1, wherein the primary test computer sends the input data to the secondary test computer using TCP/IP communication protocol, and the secondary test computer reports the exception condition to the primary test computer using TCP/IP communication protocol.
10. Programmable media containing programmable software for a primary test program that tests an application program, the programmable media containing programmable software comprising the steps of:

capturing input data entered into a primary test computer testing an application program;  
sending the input data from the primary test computer to a secondary test computer also  
testing the application program; and

receiving a report sent immediately upon detection of an exception condition to the  
primary test computer by the secondary test computer of the exception condition that occurs  
during execution of the application program by the secondary test computer responsive to the  
input data.

11. The programmable media containing programmable software of claim 10, further  
comprising the step of displaying the exception condition by the primary test computer.

12. The programmable media containing programmable software of claim 10, wherein the  
step of sending the input data to a secondary test computer includes the step of transmitting the  
input data from the primary test computer to the secondary test computer over the Internet.

13. The programmable media containing programmable software of claim 10, wherein the  
step of sending the input data to a secondary test computer includes the step of transmitting the  
input data from the primary test computer to the secondary test computer over an Intranet.

14. Programmable media containing programmable software for a secondary test program  
that tests an application program, the programmable media containing programmable software  
comprising the steps of:

receiving input data sent by a primary test computer testing an application program,

wherein the step of receiving is performed by a secondary test computer also testing the application program;

mapping the input data to a form compatible with the secondary test computer, thereby providing mapped input data, wherein the step of mapping is performed by the secondary test computer; and

when an exception condition occurs while executing the application program by the secondary test computer responsive to the mapped input data, reporting the exception condition to the primary test computer immediately upon detection of the exception condition.

15. The programmable media containing programmable software of claim 14, wherein the step of reporting the exception condition includes the step of transmitting the exception condition from the secondary test computer to the primary test computer over the Internet.

16. The programmable media containing programmable software of claim 14, wherein the step of reporting the exception condition includes the step of transmitting the exception condition from the secondary test computer to the primary test computer over an Intranet.